Noise-Aware Buffer Planning for Interconnect-Driven Floorplanning *

Shu-Min Li[1], *Yih-Huai Cherng*[2], *and Yao-Wen Chang*[3]

[1]Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan
[2]Synopsys Inc., Taipei, Taiwan
[3]Graduate Institute of Electronics Engineering & Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

Abstract

*Crosstalk-induced noise has become a key problem in interconnect optimization when technology improves, spacing diminishes, and coupling capacitance/inductance increases. Buffer insertion/sizing is one of the most effective and popular techniques to reduce interconnect delay and decouple coupling effects. It is traditionally applied to post-layout optimization. However, it is obviously infeasible to insert/size hundreds of thousands buffers during the post-layout stage when most routing regions are occupied. Therefore, it is desirable to incorporate buffer planning into floorplanning to ensure timing closure and design convergence. In this paper, we first derive formulae of buffer insertion for timing and noise optimization, and then apply the formulae to compute the feasible regions for inserting buffers to meet both timing and noise constraints. Experimental results show that our approach achieves an average success rate of 80.9% (78.2%) of nets meeting timing constraints alone (*both timing and noise constraints*) and consumes an average extra area of only 0.49% (0.66%) over the given floorplan, compared with the average success rate of 75.6% of nets meeting* timing constraints alone *and an extra area of 1.33% by the BBP method [3].*

1 Introduction

As the process technology advances into the deep submicron era, interconnect plays a dominant role in determining circuit performance and signal integrity [9]. Crosstalk-induced noise significantly affects delay and reduces signal integrity when technology improves, spacing diminishes, and coupling capacitance/inductance increases. As a result, crosstalk has become a comparably important design metric to timing, area, and power in modern VLSI design.

Buffer insertion is one of the most effective and popular techniques to reduce interconnect delay and decouple coupling effects. Traditionally, buffer insertion/sizing is performed during the post-layout stage. As the SIA technology roadmap predicts, the number of buffers inserted for performance optimization will grow dramatically [9]. It is obviously infeasible to insert hundred of thousands buffers during the post-layout stage when most routing region is occupied by routing wires. Therefore, it is desirable to incorporate buffer planning into floorplanning to ensure timing closure and design convergence.

Alpert and Devgan [1] presented formulae for computing the optimal number and locations of buffers for timing optimization. Cong, Kong, and Pan in [3, 4] and Sarkar, Sundararaman, and Koh in [6] presented pioneering works on buffer block planning for interconnect-driven floorplanning. Given a slicing floorplan, they computed the *feasible regions* for buffer insertion, clustered uniform-sized buffers into blocks, and placed these buffer blocks into the dead spaces and channels in the floorplan for timing optimization. Both [3, 4] and [6] considered an *independent feasible region* (IFR) for inserting a buffer, where the region is determined with the locations of other buffers being fixed. It has been pointed in [6] that for two-dimensional nets, IFRs are not completely independent since the assignments of buffers into their IFRs must ensure a monotonic path from the source to the sink. Further, they do not consider crosstalk noise.

For noise analysis and avoidance, there is much work in the literature. However, most of the previous works are intended for either routing or post-layout optimization. As mentioned earlier, due to the increasing design complexity, it may not be feasible to insert a large number of buffers for noise avoidance during the routing and post-layout stages. Therefore, there is a need to consider buffer planning earlier at the floorplanning stage when there is more flexibility to allocate silicon space for buffers to ensure timing closure and design convergence.

In this paper, we derive formulae of buffer insertion for both *timing* and *noise* optimization, and apply the formulae to noise-aware buffer planning for interconnect-driven floorplanning. Our algorithm is based on a three-stage approach:

- Stage 1 computes the optimal number and locations of buffers to minimize delay for a wire segment, and then computes the feasible regions that meet the timing constraints.
- Stage 2 computes the feasible regions to meet the noise constraints.
- Stage 3 finds the intersection of the feasible regions for meeting both the timing and noise constraints, allocate buffers to their feasible regions in dead space or channels, and expand a channel, if there is not enough dead space for buffer insertion.

Experimental results show that our approach achieves an average success rate of 80.9% (78.2%) of nets meeting timing constraints alone (*both timing and noise constraints*) and consumes an average extra area of only 0.49% (0.66%) over the given floorplan, compared with the average success rate of 75.6% of nets meeting *timing constraints alone* and an extra area of 1.33% by [3, 4].

2 Preliminaries

In this section, we first give some notation, then introduce the delay and noise models considering coupling capacitance.

- $\chi(u, v)$: the noise on the wire segment between two neighboring buffers u and v.
- M_v: the noise margin for a buffer or a sink v. It is the maximum allowable noise without creating any logic error. We use 0.8V, same as [2].
- k: the number of buffers to minimize the delay of a wire.
- D_{opt}: the minimum delay from source to sink with the optimal k buffers inserted.
- x: the optimal length of a wire segment between the source and the first buffer to minimize delay.
- y: the optimal length of a wire segment between every pair of neighboring buffers to minimize delay.
- Δ_i: the maximum distance that the i_{th} buffer can be moved away from its optimum location without violating the timing constraint.
- λ: the fixed ratio of coupling to total wire capacitance. We use $\lambda = 0.7$, same as [2].
- p: the slope (i.e., power supply voltage over input rise time) of all aggressor nets' signals. We use $p = 7.2$, same as [2].
- B_i: the position of i-th inserted buffer.
- Φ_i^d (Φ_i^n): the feasible region for the i-th buffer to satisfy the timing (noise) constraint.
- Φ_i: the feasible region for the i-th buffer to satisfy both the timing and noise constraints, $\Phi_i = \Phi_i^d \cap \Phi_i^n$.
- $W(\Phi_i^d)$ ($W(\Phi_i^n)$): the width of Φ_i^d (Φ_i^n).

Para.	Description (unit)	Value
r_0	sheet resistance of a wire ($\Omega/\mu m$)	0.075
c_0	unit-length capacitance of a wire ($fF/\mu m$)	0.118
T_{in}	intrinsic delay for a buffer (ps)	36.4
C_L	load capacitance (fF)	23.4
R_d	driver resistance (Ω)	180.0
c_b	input capacitance of a minimum size buffer (fF)	23.4
r_b	output resistance of a minimum size buffer (Ω)	180.0

Table 1: Parameters of the 0.18μm technology in NTRS'97.

*The work of Yao-Wen Chang was partially supported by the National Science Council of Taiwan ROC under Grant No. NSC-89-2215-E-009-117.

Throughout the rest of this paper, we will use the notation/parameters listed in Table 1 to facilitate our technical discussions. The set of parameters are based on the $0.18\mu m$ technology given in the NTRS'97 roadmap [8].

We use a switch-level RC model with an intrinsic buffer delay for a buffer, the π-model for a wire segment, and the Elmore model [5] with fringing and coulping capacitance considerations to compute the delay. See Figure 1 for an illustration.

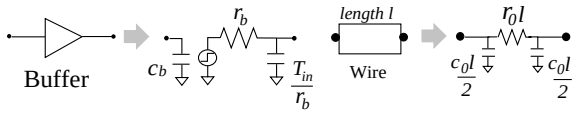

Figure 1: Switch-level RC circuits for buffers and wires.

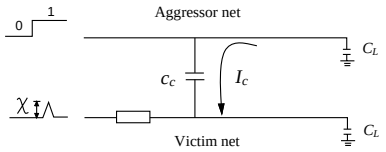

Figure 2: Noise due to crosstalk-induced current.

Figure 2 shows our noise model which considers coupling capacitance c_c. The coupling capacitance is proportional to the fringing capacitance (c_f) and the coupling length (l_c), and it is inversely proportional to the distance (d) between the aggressor and the victim nets, i.e., $c_c \propto c_f l_c/d$. We adopt a more conservative approach and set d to be the minimum wire distance d_{min} specified by the design rule. Furthermore, we set the coupling length (l_c) to be two times of the victim net's length (l_{vi}), which is the most pessimistic situation when the victim net is fully coupled from both sides by two aggressors' nets. As a result,

$$c_c = \frac{2 l_{vi} c_f}{d_{min}}. \tag{1}$$

Further, we adopt $2c_c$ in our model to express the worst case that all the aggressor nets have a different voltage transition with the victim net, for which the Miller effect makes the coupling phenomenon more significant (doubles the coupling effect). It will be clear in Section 5 that even with such most pessimistic estimation of crosstalk, our algorithms still outperform previous works. Note that our algorithms readily apply to other models with less pessimistic estimation.

Consider a wire $e = (u, v)$, where u and v are two nodes in a buffered tree. Let the length of the wire segment e be l_e, and $T(v)$ be the subtree rooted at v. $I_{T(v)}$ is the total downstream current seen at v and is the current induced by aggressor nets on downstream wires of v. The current on a unit-length wire induced by aggressor nets is $i_0 = \lambda p c_0$ [2], where c_0 is the unit-length wire capacitance, and c_c is modeled also as some fraction of the unit-length wire capacitance of the victim net. The resulting noise $\chi(u, v)$ induced from the coupling current is the voltage pulse coupled from aggressor nets in the victim net for a wire segment $e = (u, v)$. We have the following noise constraint:

$$\chi(u, v) = r_b I_{T(v)} + r_0 l_e \left(\frac{i_0 l_e}{2} + I_{T(v)} \right) \leq M_v. \tag{2}$$

The feasible region for buffers that satisfies the noise constraint [2] is given by

$$W(\Phi_i^n) \leq \sqrt{\left(\frac{r_b}{r_0}\right)^2 + \left(\frac{I_{T(v)}}{i_0}\right)^2 + \frac{2M_v}{i_0 r_0}} - \frac{r_b}{r_0} - \frac{I_{T(v)}}{i_0}. \tag{3}$$

Based on the delay and noise models, we first compute the respective feasible regions Φ_i^d and Φ_i^n for inserting buffer i to satisfy the delay and noise constraints, and then find the intersection of Φ_i^d and Φ_i^n to derive the feasible region for buffer i that meets both the delay and noise constraints. See Figure 3 for an illustration.

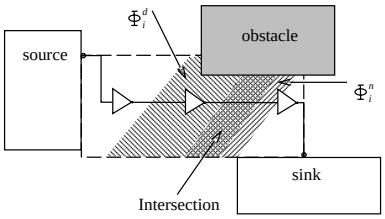

Figure 3: The respective feasible regions Φ_i^n, Φ_i^d, and $\Phi_i^n \cap \Phi_i^d$ for inserting a buffer that meet the delay, noise, and both delay and noise constraints.

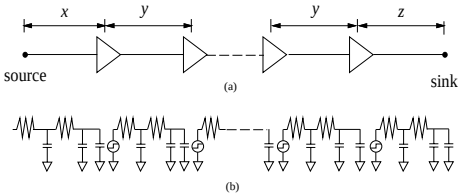

Figure 4: (a) The buffer placement: x is the optimized length between the source node and the first buffer; y is the optimized length between every pair of neighboring buffers; and z is the length between the last inserted buffer and the sink node. (b) The corresponding buffer model and wire (π) model.

3 Noise-Aware Floorplanning

3.1 Feasible Region for Delay

The feasible region for delay, Φ_i^d, is the maximum region satisfying the timing constraint T_{req} [3] for buffer i. Consider a single wire with the optimal number of k buffers inserted [1], and let the optimal delay be D_{opt}. Suppose that the i-th buffer ($1 \leq i \leq k$) is moved away from its optimum position by a distance Δ_i, where Δ_i is positive if the buffer is moved forward, and negative otherwise. Considering the worst-case coupling capacitance, in which the unit length wire capacitance become $c_0 + 2c_c$, and placing buffers at equidistance (i.e., $x = y = z$ as shown in Figure4), the following theorem specifies a sufficient condition for placing a buffer in a region that meets the timing constraint.

Theorem 1 *A solution for Δ_i ($1 \leq i \leq k$) is feasible without violating the timing constraint T_{req} if*

$$\sum_{i=1}^{k} \Delta_i^2 - \sum_{i=1}^{k-1} \Delta_i \Delta_{i+1} \leq \frac{T_{req} - D_{opt}}{r_0(c_0 + 2c_c)}. \tag{4}$$

Proof Sketch: *Let $c_w = c_0 + 2c_c$. Based on the Elmore delay model [5], the optimum delay formula is given by*

$$\begin{aligned} D_{opt} &= R_d(c_b + x c_w) + \frac{1}{2} x^2 r_0 c_w \\ &\quad + \sum_{i=1}^{k-1} \left(T_{in} + r_b(c_b + y c_w) + \frac{1}{2} y^2 r_0 c_w \right) \\ &\quad + T_{in} + r_b(C_L + z c_w) + \frac{1}{2} z^2 r_0 c_w. \end{aligned} \tag{5}$$

After inserting k buffers, we have

$$\begin{aligned} D'_{opt} &= R_d(c_b + (x + \Delta_1) c_w) + \frac{1}{2}(x + \Delta_1)^2 r_0 c_w \\ &\quad + \sum_{i=1}^{k-1} (T_{in} + r_b(c_b + (y - \Delta_i + \Delta_{i+1}) c_w) \\ &\quad + \frac{1}{2}(y - \Delta_i + \Delta_{i+1})^2 r_0 c_w) \\ &\quad + T_{in} + r_b(C_L + (z - \Delta_k) c_w) \\ &\quad + \frac{1}{2}(z - \Delta_k)^2 r_0 c_w. \end{aligned} \tag{6}$$

Note that in most cases the source and the sink are also buffers for the regeneration of signals, which implies $R_d = r_b$ and $C_L = c_b$. With this simplification, it is easy to see that the linear terms in both Equation (5)

and Equation (6) are equal when they are summed up. The difference comes from the quadratic terms. As a result, we have:

$$D'_{opt} - D_{opt} = \left(\sum_{i=1}^{k} \Delta_i^2 - \sum_{i=1}^{k-1} \Delta_i \Delta_{i+1} \right) r_0 c_w + ((x-y)\Delta_1 + (y-z)\Delta_k)\, r_0 c_w. \quad (7)$$

Placing buffers at equidistance, i.e., $x = y = z$ as shown in Figure4), we have:

$$D'_{opt} - D_{opt} = r_0(c_0 + 2c_c) \left(\sum_{i=1}^{k} \Delta_i^2 - \sum_{i=1}^{k-1} \Delta_i \Delta_{i+1} \right). \quad (8)$$

The theorem thus follows.

□

To achieve the maximal feasible region, we derive the following two corollaries demonstrating how to apply the concept of degree of freedom.

Corollary 1.1 *If a buffer can shift either forwards (positive sign) or backwards (negative sign), the maximal feasible region Δ for a buffer is given by:*

$$\Delta = \sqrt{\frac{T_{req} - D_{opt}}{(2k-1) r_0 (c_0 + 2c_c)}}. \quad (9)$$

Proof Sketch: *In the theorem we have $\Delta_i \leq \Delta$ for all $1 \leq i \leq k$. In the worst case, we may have all even-numbered buffers moved in one direction while all odd-numbered buffers moved in the opposite direction. Therefore, $\Delta_i \Delta_{i+1} \leq 0$*

By Equation (4), a solution for the maximal Δ is feasible without violating the timing constraint T_{req} if

$$\sum_{i=1}^{k} \Delta^2 + \sum_{i=1}^{k-1} \Delta^2 \leq \frac{T_{req} - D_{opt}}{r_0(c_0 + 2c_c)}. \quad (10)$$

Simplify the above equation and the condition given in the theorem becomes:

$$(2k-1)\Delta^2 \leq \frac{T_{req} - D_{opt}}{r_0(c_0 + 2c_c)}.$$

□

If all buffers can be shifted along only the same direction, the constant multiple changes from $(2k-1)$ to k. We have the following corollary.

Corollary 1.2 *If all buffers can be shifted in only the same direction, the maximal feasible region Δ for a buffer is given by*

$$\Delta = \sqrt{\frac{T_{req} - D_{opt}}{k r_0 (c_0 + 2c_c)}}. \quad (11)$$

It should be noted that a larger feasible region can be obtained under more restricted conditions. For example, if only r out of the k buffers are moved, we may replace k with r in the above two corollaries and get a larger feasible region. When we have $r = 1$, that is, only one buffer is moved, both corollaries lead to the solution:

$$\Delta = \sqrt{\frac{T_{req} - D_{opt}}{r_0(c_0 + 2c_c)}}.$$

The same result can be obtained if in Corollary 2 all buffers are moved with the same distance in the same direction.

3.2 Feasible Region for Noise

The Feasible region for noise Φ_i^n is the maximum allowable length in each net satisfying the noise margins before buffer insertion. To estimate the feasible region for noise Φ_i^n, we apply the noise formulae [2] as listed below.

The induced noise current on wire segment $e = (u, v)$ is computed by $I_e = i_0 l_e$. To satisfy the noise constraint, a buffer can be inserted at u in Equation (2), where the range of feasible region $W(\Phi_i^n)$ for buffers satisfying the noise constraint is computed from Equation (3).

$$\chi(u,v) = r_b I_{T(u)} + r_0 W(\Phi_i^n) \left(\frac{i_0 W(\Phi_i^n)}{2} + I_{T(v)} \right) \leq M_v. \quad (12)$$

We adopt 2-terminal nets as inputs, and the approach is to scan from the sink s_i with the given $M(s_i)$ to the s_0. Since the accumulated crosstalk-induced current $I_{T(v)}$ is zero for 2-terminal pins, the noise formula is modified to:

$$\begin{aligned} \chi(u,v) &= r_b I_{T(u)} + r_0 W(\Phi_i^n) \left(\frac{I_e}{2} + I_{T(v)} \right) \\ &= r_b (I_e + I_{T(v)}) + r_0 W(\Phi_i^n) \left(\frac{I_e}{2} + I_{T(v)} \right) \\ &= r_b I_e + r_0 W(\Phi_i^n) \frac{I_e}{2}. \end{aligned} \quad (13)$$

According to the above formulae (12) (13) of $W(\Phi_i^n)$, we have the feasible region for noise Φ_i^n:

$$W(\Phi_i^n) \leq \sqrt{\left(\frac{r_b}{r_0}\right)^2 + \frac{2M_v}{i_0 r_0}} - \frac{r_b}{r_0}. \quad (14)$$

In the above Equation of Φ_i^n (14), the Φ_i^n is the maximum length from next buffer B_{i+1} back to B_i without resulting in logic error. Figure 3 shows the intersection area of Φ_i^d and Φ_i^n. This intersection for each buffer guarantees to meet both the timing constraint and the noise bound.

4 Algorithm

Algorithm : Noise-Aware Buffer Insertion for Interconnect-Driven Floorplanning (NBF)
Input : A floorplan F and a set of 2-pin nets, noise constraints and timing constraints
Output : For each net, compute
(1) k—the number of buffers inserted
(2) B_i—the length between the source and the $i-th$ buffer

```
1  Construct the data structure of the horizontal and vertical polar
   graphs (HPG, VPG) for input F;
2  For each net_i
3    if (net_i violates the delay constraint)
4      Insert buffers to satisfy timing constraint; //Stage 1
5      if  (net_i's timing is satisfied by buffer insertion)
6        Apply buffer movement or additional buffer
         insertion to satisfy the noise constraint; //stage2
7        if (both noise and delay constraints are satisfied)
8          IntersectBothFRs(); //Stage 3
9          AllocateBuffers();
10   else //net_i's timing OK
11     Insert buffer to satisfy the noise constraint; //Stage 2
12     if  (both noise and delay constraints are satisfied)
13       IntersectBothFRs(); //Stage 3
14       AllocateBuffers();
15     Recheck the timing constraint;
```

Figure 5: The Noise-aware Buffer insertion for interconnect-driven Floorplanning (NBF) algorithm.

Circuit	# modules	# nets	# pads	# 2-pin nets
a9c3	147	1202	22	1613
ac3	27	212	75	446
ami33	33	123	43	363
ami49	49	408	22	545
apte	9	97	73	172
hc7	77	449	51	1450
hp	11	83	45	226
playout	62	2506	192	2150
xc5	50	1005	2	2275
xerox	10	203	2	455

Table 2: Statistics of the MCNC benchmark circuits.

The NBF algorithm shown in Figure 5 is a three-stage approach:

	#Net/%Net Meet			#Buffers Inserted			Area Expansion Ratio			CPU Time (sec)		
Circuit	**BBP**	**NBF$_d$**	**NBF$_{d\&n}$**	**BBP**	**NBF$_d$**	**NBF$_{d\&n}$**	**BBP**	**NBF$_d$**	**NBF$_{d\&n}$**	**BBP**	**NBF$_d$**	**NBF$_{d\&n}$**
a9c3	1450/89.9%	1421/88.1%	1421/88.1%	4316	1093	2250	1.05%	0.04%	0.08%	4.36	20.56	12.63
ac3	366/82.1%	363/81.4%	346/77.6%	718	383	479	1.39%	0.03%	0.19%	0.21	1.58	1.53
ami33	302/83.2%	251/69.1%	230/63.4%	703	617	399	0.93%	2.84%	2.74%	0.22	1.89	2.58
ami49	398/73.0%	525/96.3%	525/96.3%	949	1716	1776	0.65%	0.08%	0.33%	0.46	4.46	4.71
apte	132/76.7%	94/54.7%	90/52.3%	262	108	92	1.44%	0.39%	0.47%	0.06	0.49	0.52
hc7	1079/74.4%	1315/90.7%	1305/90.0%	2847	1066	1342	2.00%	0.05%	0.6%	2.18	5.98	6.34
hp	154/68.1%	175/77.4%	175/77.4%	301	481	481	1.05%	0.25%	0.96%	0.08	0.6	0.69
playout	1478/68.7%	1977/92.0%	1763/82.0%	4262	895	1377	0.71%	0.06%	0.19%	1.95	26.16	30.55
xc5	1665/73.2%	2132/93.7%	2093/92.0%	2941	708	983	2.69%	0.22%	0.63%	1.28	9.97	10.16
xerox	304/66.8%	296/65.1%	287/63.1%	519	542	480	1.39%	0.91%	0.39%	0.1	1.63	1.67
Average	733/75.6%	784/80.9%	758/78.2%	1782	761	966	1.33%	0.49%	0.66%	1.09	7.33	7.14

Table 3: Experimental results on the MCNC93 benchmark circuits (Both NBF$_d$ and NBF$_{d\&n}$ ran on a 400 MHz SUN Ultra 60 workstation, while BBP ran on a 500 MHz Intel Pentium-III PC).

- Stage 1: As shown in Figure 4(a), we compute Φ_i^d. For each net, compute the optimal number k, and the distance x, y, and z. Also we have D_{opt} from the source s_0 to the sink. For each buffer inserted in this net, compute the buffer position B_i and its feasible region for timing Φ_i^d.
- Stage 2: Illustrated in Figure 6, we use both buffer movement and buffer insertion to meet the noise constraint. For each buffer location B_i, from B_k back to B_1, we find Φ_i^d, Φ_i^n, and the intersection (Φ_i) of Φ_i^d and Φ_i^n. There are three possible relations between Φ_i^d and Φ_i^n as follows.
  - Case 1 (with buffer insertion): There is no intersection between Φ_i^n and Φ_i^d. We apply buffer insertion to meet the noise constraint (see Figure 6(a)).
  - Case 2 (with buffer movement): This case occurs when Φ_i^d and Φ_i^n overlap, but the buffer does not located in the intersection region. Thus, we move the i-th buffer to the left end of Φ_i^n by the distance $l_i - W(\Phi_i^n)$, where l_i is the distance between the i-th and the $i+1$-th buffers. See Figure 6(b).
  - Case 3 (without buffer insertion or movement): This case occurs when the delay-optimal position of the i-th buffer is located in the intersection of Φ_i^d and Φ_i^n. For this case, the i-th buffer remains at its original optimal position B_i. See Figure 6(c). Furthermore, this implies that there exists a noise slack of $W(\Phi_i^n) - l_i$ (in terms of the distance).
- Stage 3: The third stage is to allocate buffers to Φ_i. We allocate buffers to their feasible regions in dead space or channels within the routing region. If there is not enough capacity in dead space and channels, we will expand a channel. In this case, the height and/or the width of the whole chip is expanded and the extra area is computed accordingly. Note that we must recompute the timing and noise for all nets involved in the expanded channel to check if both constraints are still satisfied.

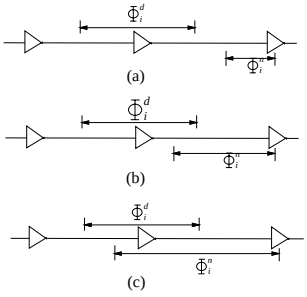

Figure 6: Three cases for the intersection of Φ_i^d and Φ_i^n.

5 Experimental Results

The NBF algorithm was implemented in the C language on a 400 MHz Sun Ultra 60 workstation and experimented on the MCNC benchmark circuits used in [3, 4, 6] (see Table 2 for the circuits and their statistics). We compared NBF$_{d\&n}$ (the NBF algorithm with both delay and noise considerations) and NBF$_d$ (the NBF algorithm with delay optimization alone) with the buffer block planning algorithms BBP presented in [3, 4] based on the same data, initial floorplans, delay budgets and parameters generated in [3, 4].

The parameters for interconnects and buffers are given in Table 1. All parameters used here are the same as [2, 3, 4]. Same as [2], we adopted the rise time for an aggressor net as 0.25 ns, the power supply voltage as 1.8 V, and noise margin as 0.8 V. Same as [3, 4], all nets were 2-pin connections and the power/ground nets were excluded. Also, we used the delay constraints provided by the authors of [3, 4], which were originally randomly generated in $[1.05T_{opt}, 1.20T_{opt}]$. (Note that we did not compare with [6] because it used different constraints.)

Table 3 gives the results obtained from the BBP, NBF$_d$, and NBF$_{d\&n}$ algorithms. The columns "#Net%Net Meet" give the number/percentage of the nets that meet the delay constraints alone for BBP and NBF$_d$, and *both the delay and noise* constraints for NBF$_{d\&n}$. The columns "#Buffers Inserted" list the total numbers of buffers inserted to satisfy the given constraints. The columns "Area Expansion Ratio" report the percentages of areas increased after buffer insertion, computed by $(the\ new\ chip\ area - the\ original\ chip\ area)/the\ original\ chip\ area$. The columns "CPU Time" give the running times in second for the algorithms.

As shown in Table 3, NBF$_d$ (NBF$_{d\&n}$) achieves an average success rate of 80.8% (78.2%) of nets meeting the timing (both timing and noise) constraints, uses only 761 (966) buffers in total, and consumes an average extra area of only 0.49% (0.66%) over the given floorplan, compared with the average success rate of 75.6% meeting timing constraints, 1782 buffers inserted, and extra area of 1.33% resulted from BBP. The experimental results show that NBF$_d$ and NBF$_{d\&n}$ perform much better than BBP, and the overhead for NBF$_{d\&n}$ to meet the additional noise constraints is quite small (2.6% fewer nets meeting the constraints and a 0.17% larger area). Further, our algorithms tend to obtain large gains for larger circuits; e.g., for the largest circuit xc5, NBF$_d$ (NBF$_{d\&n}$) achieves an average success rate of 93.7% (92.0%) of nets meeting the timing (both timing and noise) constraints while BBP obtains the average success rate of only 73.2% of nets meeting the timing constraint alone. The results reveal that our feasible region formulae for both delay and noise are very effective in improving multi-metric performance by consuming very small resources. The running times for our algorithms are reasonable. (Note that the CPU times reported in Table 3 are based on different machines.)

References

[1] C. J. Alpert and A. Devgan, "Wire Segmenting for Improved buffer insertion," *Proc. DAC*, pp. 588–593, 1997.

[2] C. J. Alpert and A. Devgan, S. T. Quay, "Buffer Insertion for Noise and Delay Optimization," *IEEE Trans. CAD*, Vol. 18, Issue 11, pp. 1633–1645, Nov. 1999.

[3] J. Cong, T. Kong and D. Z. Pan, "Buffer Block Planning for Interconnect-Driven Floorplanning," *Proc. ICCAD*, pp. 358–363, 1999.

[4] J. Cong, T. Kong and D. Z. Pan, "Buffer block planning for interconnect planning and predicition," *IEEE Trans. on VLSI Systems*, 2001.

[5] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide band amplifiers," *J. Appl. Phys.*, vol. 19, pp. 55–63, 1948.

Proc. ICCAD, pp. 726–731, 1997.

[6] P. Sarkar and C.K. Koh, "Routability-Driven Repeater Block Planning for Interconnect-Centric Floorplanning," *IEEE Trans. on Computer-Aided Design*, vol. 20, no. 5, pp. 660–671, May 2001.

[7] X. Tang and D. F. Wong, "Planning buffer locations by network flows," *Proc. ISPD*, pp. 180–185, April 2000.

[8] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*, 1997.

[9] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors 1999 Edition*, Nov. 1999.